J. Voldman
Lee W. Hoevel

The Software-Cache Connection

This paper describes an adaptation of standard Fourier analysis techniques to the study of software-cache interactions. The cache is viewed as a "black box" boolean signal generator, where "ones" correspond to cache misses and "zeros" correspond to cache hits. The spectrum of this time sequence is used to study the dynamic characteristics of complex systems and workloads with minimal a priori *knowledge of their internal organization. Line spectra identify tight loops accessing regular data structures, while the overall spectral density reveals the general structure of instruction localities.*

1. Introduction

Recent improvements in machine organization have reduced the number of processor cycles required to execute individual instructions in high performance systems. The cycle time itself has also been reduced by advances in chip and packaging technology. Main storage access time has not kept pace with these two trends, however, but has remained almost constant for several years.

This has resulted in the so-called *finite cache problem*—a situation where efforts to increase system performance are blocked by a bottleneck between the cache and its backing store. For the foreseeable future, this problem will be so severe that the per-instruction penalty imposed by a finite cache may equal or even exceed the average per-instruction execution time: *i.e.*, half the potential performance of current and projected systems may be lost within electronic levels of the memory hierarchy.

Traditional solutions to this problem involve "tuning" various parameters of cache and processor organization—such as cache size, cache associativity, cache line size, and processor pipeline depth—to the expected work load. These hardware alternatives have been studied extensively, and a substantial body of data exists describing their relation to finite cache penalty and work load. Unfortunately, the technological trends noted above leave little hope that this conventional methodology will continue to produce satisfactory results.

The intent of modern processor design in this regard is to minimize the delay caused by short range data and address interlocks—no more than three or four references. Our objective is to determine whether or not there is any predictability to cache misses, especially at long ranges. Such a capability would enable us to improve cache performance through intelligent cache management (*replacement*) or anticipatory line transfers (*prefetching*).

Long-range prediction would be of particular value in constructing efficient dispatch and queueing strategies, given the volatility of buffer miss ratio in contemporary systems. For example, Fig. 1 shows how the buffer miss ratio changes over time for a typical data base work load and cache organization. The horizontal axis calibrates time in terms of storage references, while the vertical axis calibrates the buffer miss ratio over 100 references. The degree of variation evident in this graph is characteristic of such complex environments and cannot be explained satisfactorily in terms of physical cache design parameters alone.

We believe that software related events are the primary cause of significant fluctuations in the buffer miss ratio, and hence that the key to improving cache performance lies in understanding the specific software mechanisms involved. This paper reports our initial experimental attempts to substantiate and quantify this intuitive thesis.

Copyright 1981 by International Business Machines Corporation. Copying is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract may be used without further permission in computer-based and other information-service systems. Permission to *republish* other excerpts should be obtained from the Editor.

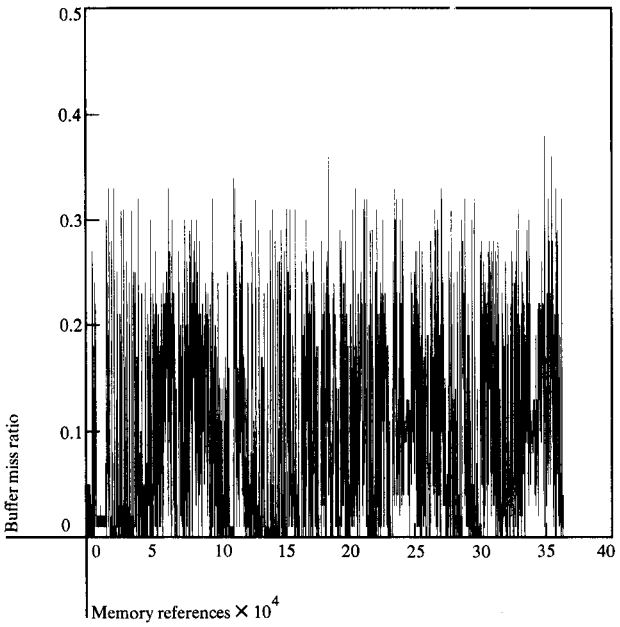

Figure 1 Buffer miss ratio computed every 100 references.

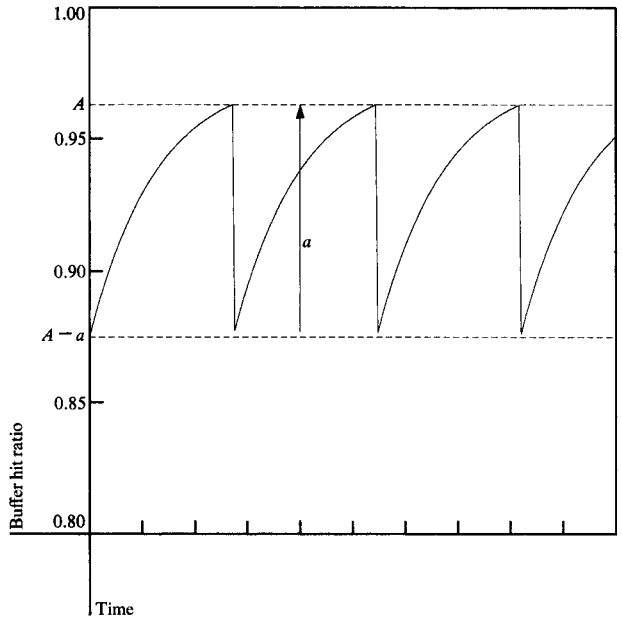

Figure 2 The capacitor model.

2. The need for a closer look

We believe that a substantial fraction of the cache misses that occur in modern systems are due to dynamic changes in locality [1, 2]. Task switches are a clear indication of such changes in locality, and intuition suggests the following informal relation between cache misses and task switches. Let A be the "steady state" value of the buffer hit ratio (*bhr*) in a multitasking environment. Each time a task switch occurs, we expect the sudden change in locality of references to cause $bhr(t)$ to drop from a steady state level A to some smaller value $A - a$. Thereafter, $bhr(t)$ should climb slowly back to the steady state level A as the new locality becomes cache resident, like the rise in voltage drop across a charging capacitor (Fig. 2).

This model is discussed at length in reference [3]. As will be seen, however, although this model correlates well with experimental data, it reveals only part of the true picture. Consider, for example, the variations in buffer miss ratio over time shown in Fig. 1. There are over 150 task switches spread across the span of 360 000 references covered by the horizontal (time) axis. The above-noted intuitive model leads us to expect a sharp peak after each pass through the dispatcher, followed by an exponential decay back to a residual steady state.

Obviously, the phenomenon is much more complex, and apparently more random, than anticipated. Not only is it difficult to pick out specific instances of the "cache capacitor" charging and discharging, it is even difficult to find any obvious steady state!

In fact, the graph more closely resembles a plot of sunspot activity, or perhaps a very noisy and poorly grounded signal displayed on an oscilloscope. This analogy prompted the idea of using signal analysis techniques to try to separate the noise from the signal—assuming, of course, that there is indeed any signal there.

Focusing on buffer miss ratio alone may be too simplistic, since it is only a one parameter measure of the cumulative effects of the mechanisms involved here. Our thought is that the spectrum for the *bmr* will help separate the dynamic effects of these various mechanisms, revealing periodic signals hidden in the visual noise of Fig. 1 [4, 5].

3. Representing the buffer miss ratio

Given an interval of time, the buffer miss ratio can be calculated by dividing the number of misses occurring in the interval by the total number of references occurring in the interval. If we partition time into small intervals, $I(t)$, for t varying from 1 to N, then we can define a function $bmr(t)$ relating the cache misses to time as

$$\frac{M(t)}{(M(t) + H(t))},$$

where $M(t)$ is the number of cache misses in interval t,

and $H(t)$ is the number of cache hits in interval t. The question of representation is: How long should each interval be?

Clearly, if too large an interval is selected, high frequency information will be lost, or at least severely attenuated. If too short an interval is selected, then there will be large variations in the buffer miss ratio from one interval to the next, and smoothing the data will be a problem.

We decided to deal with the smallest interval size possible (one reference), in which case $bmr(t) = M(t)$. Thus, we define the *boolean miss sequence* $M(t)$ as the series of zeros and ones representing the net effect of passing a given reference stream through a given cache organization. This phenomenon is represented in $M(t)$ by reflecting each cache hit as a zero and each cache miss as a one. Clearly, $M(t)$ will vary depending on cache organization and becomes well defined only when a particular cache is specified.

To construct $M(t)$ at this level of detail, we first process a raw instruction trace tape using a machine simulator to generate a corresponding reference sequence R, where $R(t)$ is the address of the tth reference to storage produced by the machine simulator. This reference sequence is then supplied as input to a cache simulator (which approximates the cache on a System/370 Model 168-3) that produces the desired miss sequence. In other words, $M(t) = 0$ if $R(t)$ results in a cache hit, and $M(t) = 1$ if $R(t)$ results in a cache miss when processed by a given cache simulator.

4. Mathematical background

There is little need for an in-depth review of the mathematics and techniques associated with discrete Fourier transforms in this paper, given the number of excellent and widely available texts in this area [6–10]. However, it may prove helpful to present the basic equations as applied to our problem.

Let $M(t)$ be a function relating cache misses and time. Then

$$F(\omega) = \int_{-\infty}^{+\infty} M(t)e^{-i\omega t}dt$$

is the Fourier transform of $M(t)$,

$$S(\omega) = F(\omega) \times F^*(\omega) = |F(\omega)|^2$$

is the spectrum of $M(t)$, and

$$\rho(\tau) = \overline{M(\tau) \times M(\tau - t)} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(\omega)e^{i\omega\tau}d\omega$$

is the autocorrelation function of $M(t)$. In particular, for $\tau = 0$, we have

$$\rho(0) = \text{variance} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} S(\omega)d\omega.$$

Mathematically, then, $S(\omega)$ describes how the variance of $M(t)$ is distributed over the frequency domain.

However, we have chosen $M(t)$ to be a binary sequence, and we therefore can write

$$\overline{M(t)} = M^2(t)$$

and

$$\overline{M(t)} = \frac{1}{N} \sum \text{“ones.”}$$

Thus,

$$\overline{M(t)} = \frac{number \quad misses}{N} = bmr,$$

bmr denoting the classical buffer (cache) miss ratio as we know it.

Then, the variance of $M(t)$ will be

$$V[M(t)] = \overline{[M(t) - bmr]^2} = \overline{M^2(t)} - bmr^2,$$

which can be rewritten as

$$V[M(t)] = bmr - bmr^2 = bmr(1 - bmr)$$

and

$$V[M(t)] \cong bmr.$$

Therefore, the variance of the binary sequence $M(t)$ is, at a first order approximation, the cache miss ratio. Hence one can think of the spectrum as showing how the cache miss ratio itself is distributed over the frequency domain.

5. Miss sequence spectrum

The transform of this miss sequence was computed by applying a double precision discrete fast Fourier transform routine [11]. Its spectrum (Fig. 3) was calculated by averaging the sum of the squares of the real and imaginary components of this transform over each 100 points in the frequency domain. The vertical axis is logarithmic, as is conventional in this type of analysis, while the horizontal axis is linear and calibrated in normalized frequency units from 0.0 to 0.5 cycles. The “time” in this analysis is measured in successive references, *i.e.*, first reference, second reference, etc. The normalized frequency scale (normalized to the length of the trace measured in references) is therefore expressed in occurrences per reference. Some event with frequency 1/5 happens once every

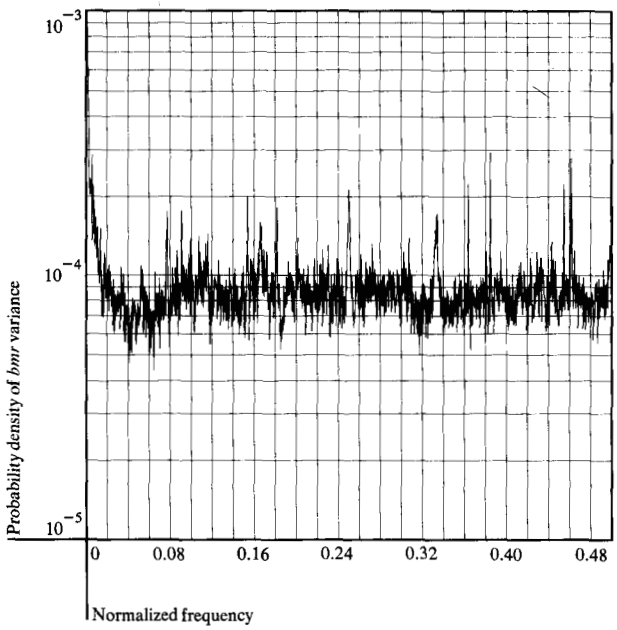

Figure 3 Miss spectrum of Trace 1.

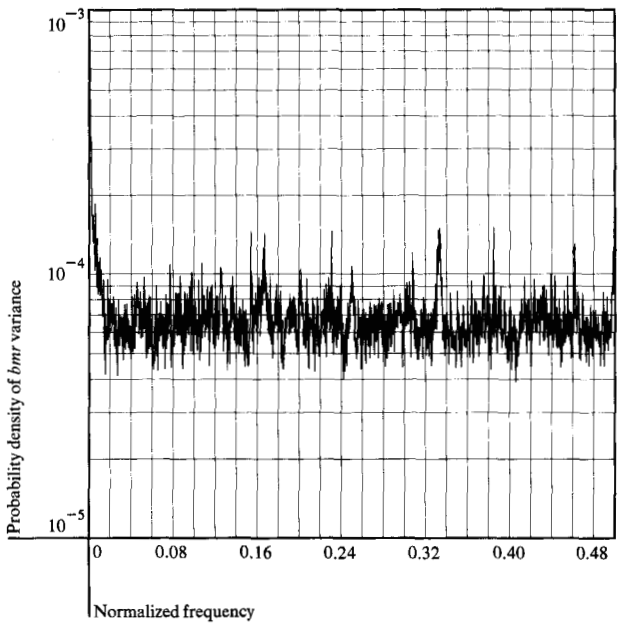

Figure 4 Miss spectrum for cache model with 64-byte line.

five references. This allows an easy comparison between spectra of traces of different lengths. Had we been in a position to use actual time, then the frequency would have been expressed in hertz; however, actual time in seconds can only be computed by assuming some specific machine and would have in this case significantly restricted the generality of the results pertaining to software by introducing delays due to a specific machine organization.

On the other hand, we think it will certainly be of value to do such an analysis using a real time scale and therefore focus on the effect of different machine organizations.

This spectrum contains many sharp spikes and is definitely not a flat line. This conveys information: Any departure of the spectrum from the shape of white noise indicates some predictable behavior in the time domain. Let us consider a few specific features of interest appearing in this spectrum.

There is a clear spike in the middle of the spectrum, at a normalized frequency of 0.25. This corresponds to a period of four references. The simulated machine was a double-word wide, so that four references access 32 bytes. This happens to be the line size of the simulated cache, and we therefore suspected that this spike was related to the cache line size. To check this experimentally, we developed a cache model with a 64-byte line and computed the Fourier transform and spectrum of its associated miss sequence in the manner described above. Figure 4 shows that the spike in the middle has decreased in amplitude, as expected.

Indeed, the amplitude of this spike is almost exactly halved. As will be seen later, this observation is a strong indication that the spike is caused by long bursts of purely sequential references. Further, the spikes at frequencies 0.125 and 0.375 are clearer on Fig. 4 than on Fig. 3 and are of about the same amplitude as the vestigial spike at 0.25.

This indicates that some sequential reference activity remains significant, even when doubling cache line size.

In changing the cache model from a 32-byte line to a 64-byte line, we noticed that the spikes at frequencies 0.0772, 0.0908, 0.1539, 0.1819, 0.2017, 0.2728, 0.3639, 0.3847, 0.4547, and 0.4617 decreased significantly. The spikes at frequencies 0.1667 and 0.3333 (its harmonic) decreased only slightly, which indicates weak sensitivity to the line size. In addition, a spike at 0.23 appeared on the spectrum associated with a 64-byte line cache that was not significantly above the noise level for the 32-byte line spectrum.

A closer look at these values shows that we have actually uncovered three harmonic series stemming from three distinct low frequency fundamentals, as shown in Table 1.

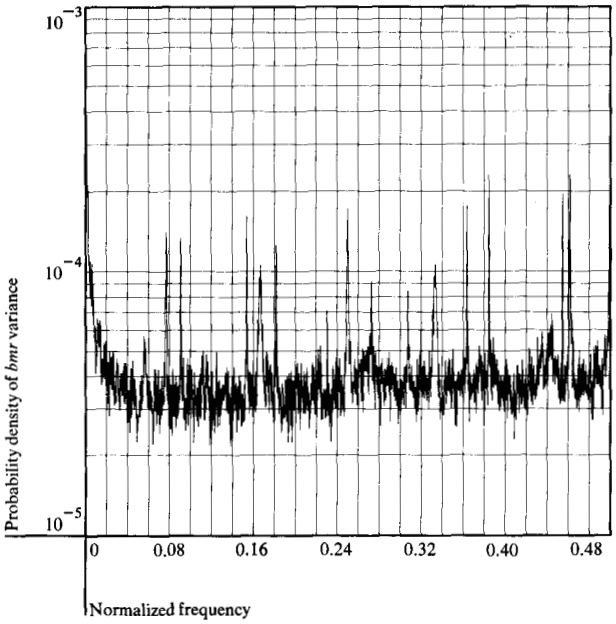

Figure 5 Data miss spectrum for Trace 1.

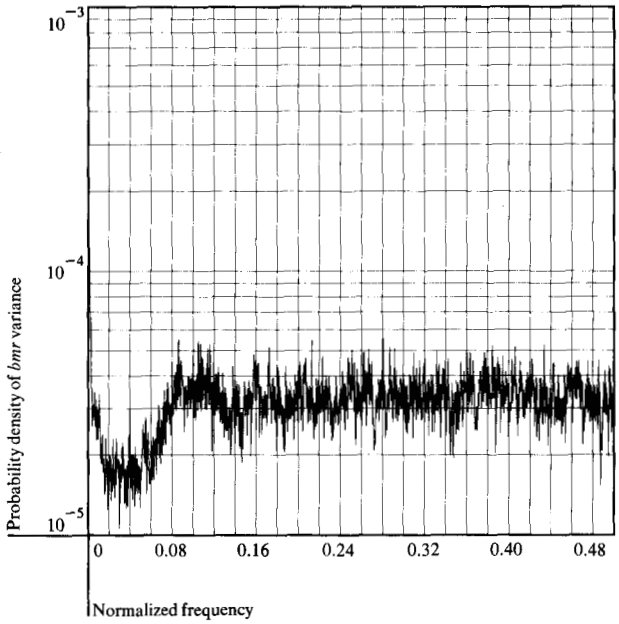

Figure 6 Instruction miss spectrum of Trace 1.

Furthermore, the extreme low frequency part of this spectrum is obviously not flat, rising far above the spectral level associated with white noise. We interpret this as evidence that low frequency events have quite predictable effects on the buffer miss ratio. Since the frequency of occurrence of software events is commensurate with this range, it should be possible to anticipate the misses triggered by certain software events.

● *Filtered sequences*

One of the advantages of spectral analysis is that it permits the use of filters to isolate events of specific frequencies. We thought it would be instructive to see what time domain events correspond to specific frequency domain spikes, and also to see what happens when the miss sequence is partitioned according to reference type—*i.e.*, separated into disjoint instruction and data miss sequences.

Frequency domain filtering

The harmonic series discussed above piqued our curiosity, and we decided to use a frequency domain filter to study them further. A special cache simulation was performed that zeroed out every one (or miss) due to an instruction fetch. The resulting data miss sequence was then converted into the spectrum shown in Fig. 5 in the usual manner. Note that again several significant, sharp spikes are readily visible.

In fact, the spectral pattern is remarkably similar to that shown in Fig. 3, which implies that most of the spikes are due to data movement as opposed to instruction fetches. This was a surprise; we had expected periodic components in the instruction sequence, but not in the data sequence. Not only does the data spectrum indicate periodic components, but the instruction spectrum (Fig. 6) is devoid of such features.

Table 1 Harmonic series.

Series	*A*	*B*	*C*
Fundamental	0.0772	0.0908	0.1667
1st harmonic	0.1536	0.1819	0.3333
2nd harmonic	0.2308	0.2728	
3rd harmonic	0.3086	0.3639	
4th harmonic	0.3847	0.4547	
5th harmonic	0.4617		

The reader familiar with filter theory knows that it would be costly in computing time and difficult in general to design a sufficiently narrow bandpass filter in the time domain, with sharp enough cutoff frequencies, to isolate only that part of the time domain signal that gives birth to a spike in the frequency domain. The advantage of using the discrete Fourier transform and a digital computer is that it is very easy to work in the frequency domain, on the transform itself.

The discrete Fourier transform of the data miss sequence was set to zero at all frequencies except those

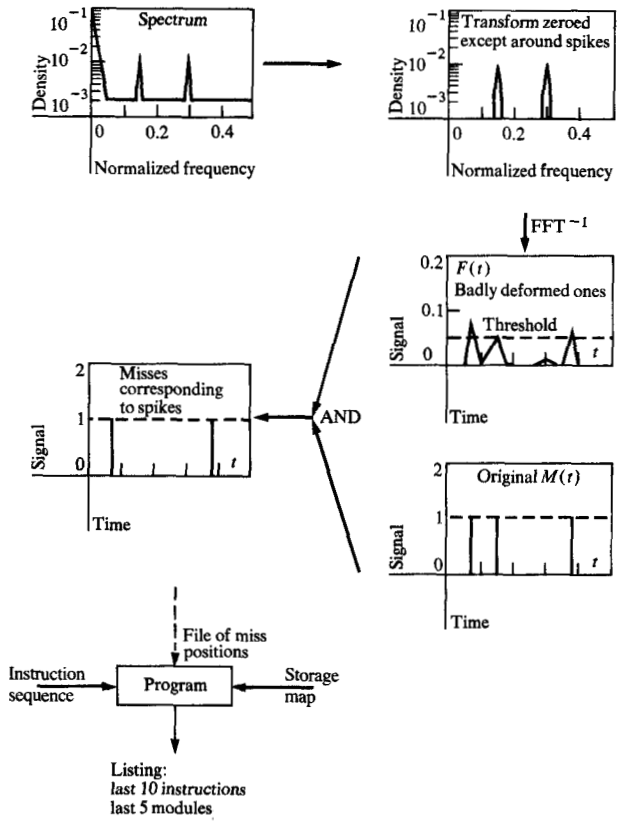

Figure 7 Procedure for generating filtered, zero-one sequence.

falling within the spikes of the A harmonic series themselves. This filtered transform was then converted back into the time domain using an inverse fast Fourier transform, resulting in a signal whose amplitude varied between 0.0 and 0.135.

After studying this filtered time domain signal, we decided to generate a filtered, zero-one sequence using the following algorithm:

> If the original data miss sequence contained a miss at a position in which the value of the filtered signal was at or above an arbitrary threshold value (0.120), then a one was generated; otherwise a zero was generated.

Our rationale was that such a high value in the filtered signal corresponded to a badly deformed one, and therefore to a filtered miss. The logical "anding" with the original sequence was used to eliminate spurious ones resulting from side effects of the narrow filtering itself.

We then produced a list of reference addresses and responsible instructions for each one remaining in the sequence of confirmed misses generated by the above filtering process.

A complete description of this entire process can be seen on Fig. 7.

We were greatly surprised to find that all 162 misses selected by this filter originate from the same instruction, a Compare Logical Immediate (CLI). Almost all of the actual addresses corresponding to these misses reference the bottom of memory and are offset exactly 32 bytes from the adjacent miss addresses. This part of the memory is known to contain a set of data structures, called Unit Control Blocks (UCBs) in system programmer jargon, which represent Input/Output devices for the operating system.

Subsequent analysis showed that the CLI instruction was part of a tight loop of ten instructions that indeed was doing a sequential search for a specific byte in a specific UCB. In addition, we found that this event (the start of a UCB lookup) only happens twice on Trace 1, both occurrences falling in the second half of the trace.

This experiment demonstrates the power of spectral analysis as a tool for discovering and dissecting software and cache problems.

Data miss sequence

Recall the two relatively large spikes appearing at frequencies 0.167 and 0.333 in Figs. 4 and 5 and listed in column C of Table 1. They correspond to respective periods of six references and three references; on the simulated machine, this is equivalent to 1.5 and 0.75 cache lines. The other very sharp spikes on the spectrum do not fall on periods that correspond to primitive multiples of either references or cache lines.

Before drawing any conclusions on spike semantics, however, we felt that it would be desirable to have an independent element of comparison—*i.e.*, an instruction trace reflecting a different work load.

Trace 2 was intended to capture typical commercial "batch" processing executed under the same operating system used to generate the Trace 1 benchmark. As before, a zero-one miss sequence was produced, processed by the fast Fourier transform, and converted into the spectrum shown in Fig. 8.

One can see five significant spikes on this picture, with the following reference periods: 8, 5, 4, 2.7, and 2.5. Again, we have two harmonic sequences: the spikes at periods 8, 4, and 2.66; and the spikes at periods 5 and 2.5.

Our colleague, Thomas R. Puzak, suggested that it might be helpful to filter out those references generated

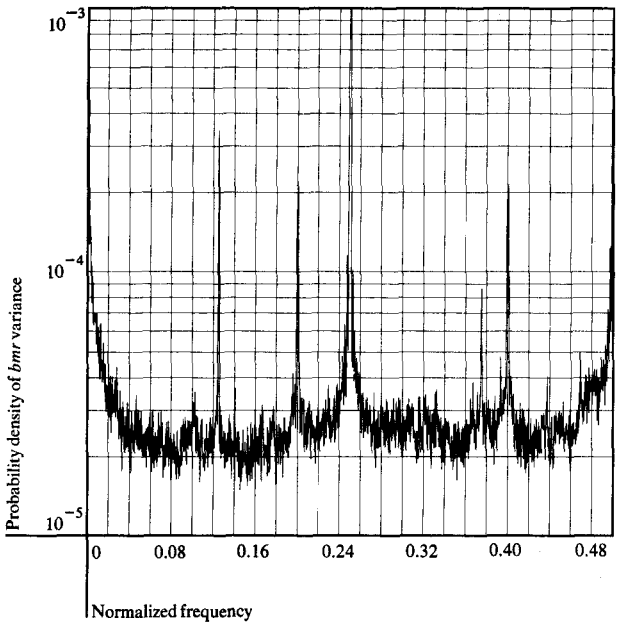

Figure 8 Data miss spectrum of Trace 2.

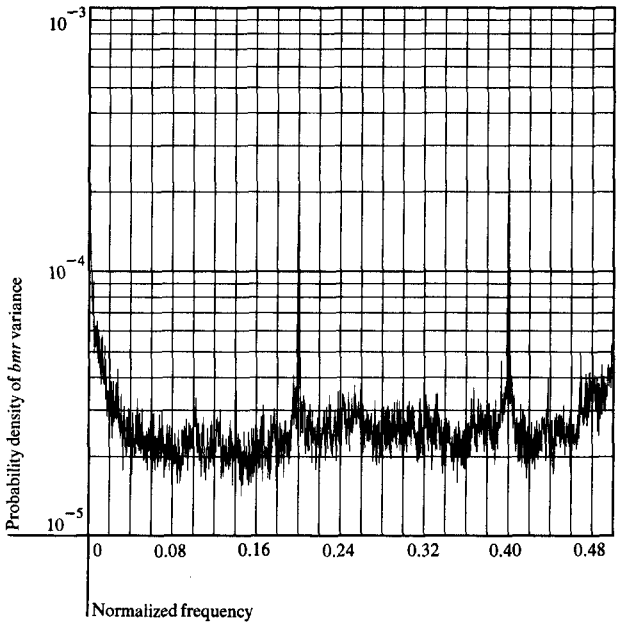

Figure 9 Data miss spectrum of Trace 2 with MVCL effects removed.

by the MVCL instruction. The cache simulator was modified to report all MVCL-based references as cache hits, and the spectrum of the resulting reference sequence is shown in Fig. 9. Observe that one of the series of harmonics (those spikes at frequencies 0.125, 0.250, and 0.375, corresponding to periods 8, 4, and 2.7) disappeared entirely, while the other was essentially unaffected. This suggests that these spikes are related to purely sequential use and movement of data. In general, we speculate that such a pattern of spikes, with periods closely related to the line size of the cache, are associated with serial processing of data and instructions (serial processing need not be sequential; random data can be accessed serially, as in control block chaining).

It is natural to question whether this spectrum is characteristic of batch processing *per se*, or only of some specific events in a particular trace sample capturing only a fraction of a second in real time. In a study made after the main results for this paper were obtained, we found that spikes do happen on a local basis. More specifically, they are related to tight loops into regular data structures, and the spectral analysis should be conducted on small sections of trace (16K references).

It is our contention that the data miss spectra (by small sections) are a general signature for the type of data reference activity occurring in a given work load, and hence can be used to discriminate between work loads that are intuitively perceived as different—even when conventional statistics fail to reveal any distinguishing characteristics.

Spectral synthesis To further our understanding of the meaning of the spectral lines in these previous experiments, we followed a suggestion from Shmuel Winograd (who has helped us avoid several pitfalls throughout this study) that we focus on the properties of certain synthetic sequences.

We first constructed a building block sequence, $m(t)$, of period 4, *i.e.*,

$m(t)$ = "100010001000 · · ·"

and of length 50 000. The spectrum of this periodic function will contain a regular pattern of spectral lines at normalized frequencies $r/4$ (for $r = 0$ to 3), as explained in Appendix A.

All of these spectral lines will have the same height, which is determined by the total number of ones in the time domain. Because the time is measured in successive references, *i.e.*, first reference, second reference, · · ·, nth reference, the frequencies will be measured in number of occurrences per reference. In other words, a normalized frequency of 0.25 (1/4) should be read as meaning one occurrence of an event every four references. The term "normalized" implies that the number of

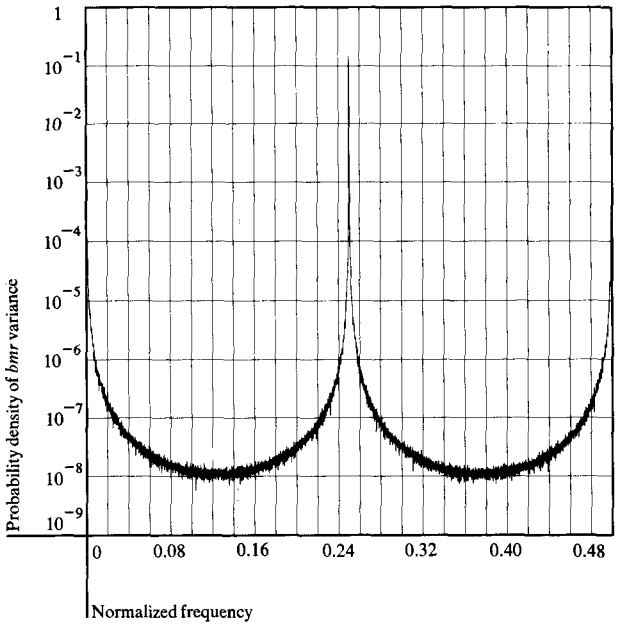

Figure 10 Spectrum for $M(t)$ with subsequences 50 000 points long with period four.

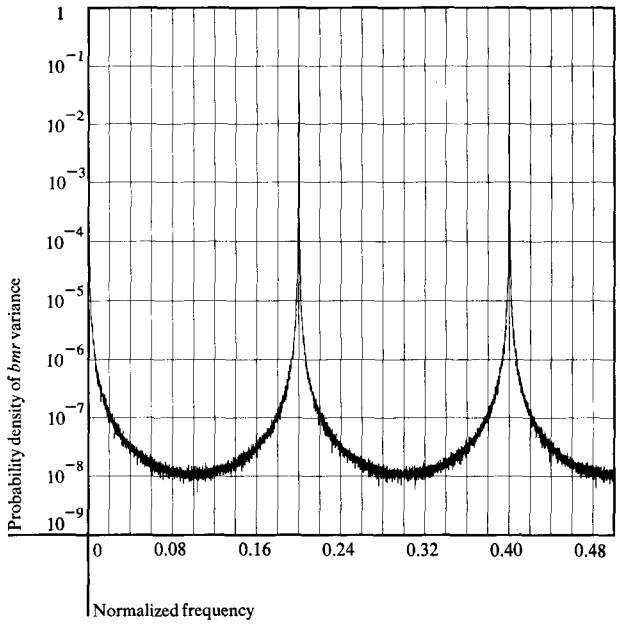

Figure 11 Spectrum for $M(t)$ with subsequences 50 000 points long with period five.

occurrences of some event on a trace (frequency) has been normalized to the length of the trace expressed in references.

We then constructed a miss sequence $M(t)$ using eight such $m(t)$ subsequences, shifted one from another by a uniformly random number of references not greater than 3000. The Fourier transform for $M(t)$ will be the sum of the Fourier transforms for each of the $m(t)$ building block sequences, with each spectral line shifted in phase by

$$e^{-ir2\pi q_i/4},$$

where q_i is δ_i modulo(4), δ_i is the relative shift for building block number i, and $r = 0, 1, 2, 3$ is the integer multiplier used to identify a particular spectral line (the harmonic number).

The spectrum for $M(t)$ is shown in Fig. 10. It is a known property of a spectrum to be symmetric about frequency 0.5. This is why we only plot a spectrum from 0 to 0.5. The periodicity of the "building block" sequences is, as expected, reflected by a spike at the fundamental frequency of 0.25. This implies that the spectral line at 0.25 on the spectrum for Trace 2 is the result of one or more occurrences of an event which manifests itself by a long (compared to the period) sequence of misses at every four references.

Indeed, we have just identified such a phenomenon in the previous section: the execution of MVCL instructions that move a significant number of bytes. This confirms the intuitive notion and general methodology described above, and holds out the hope that we may eventually be able to "read" underlying behavioral patterns in the miss spectrum of actual trace tapes.

A similar computation using 8 subsequences of internal periodicity 5 (there is nothing magic about the number 8; it ended up being the needed number to fill 360 000 points) and length 50 000 produces the spectrum shown in Fig. 11. It exhibits spikes at 0.20 and 0.40 similar to those found on the data miss spectrum of Trace 2 (Fig. 8). The sum of the spectra in Figs. 10 and 11, then, captures most of the significant features of the Trace 2 spectrum.

We thus have shown that the pattern of spectral lines in the spectrum of an actual tape can be synthesized rather easily, and that these lines can be explained as manifestations of infrequent software events of relatively long duration that exhibit a very regular miss pattern within each occurrence.

Impact of periodic sequences With so many spectral lines showing up on the data miss sequence of Trace 1, one cannot help but wonder how many misses they represent. Using the same filtering procedure described earlier, we produced a miss sequence, whose spectrum is shown in Fig. 12, with all the spectral lines filtered out.

We started with 14 450 data misses on the spectrum of Fig. 5, and we are left with 8228 misses in Fig. 12.

Therefore, all the periodic phenomena corresponding to the spectral spikes represent a total of 6222 data misses; *i.e.*, 43 percent of the total number of data misses.

Therefore, 43 percent of all data misses on this tape exhibit a very periodic and thus predictable behavior. A mechanism using this information could therefore eliminate a substantial fraction of all data misses in similar programming environments.

Instruction miss sequence

We define an instruction fetch miss sequence $M(t)$ as a series of zeros and ones that represents the net effect of passing a given reference stream through a given cache organization. If $M(t) = 1$, then the tth reference was an instruction fetch that resulted in a cache miss. If $M(t) = 0$, then the tth reference was either not an instruction fetch or resulted in a cache hit.

The cache simulation used with Trace 1 was again modified in such a way that all misses not due to pure sequential instruction fetches were reported as cache "hits" (*i.e.*, zeroed out). This resulted in an instruction miss sequence whose associated spectrum is shown in Fig. 6. It is clear that the instruction miss spectrum differs markedly from the data miss spectrum.

In particular, it has an obvious dip in the low frequencies. The frequency of the minimum is equal to 0.035 cycle (a period of 28.5 references). Between 0.5 and 0.08 cycle (periods between 2 and 12.5 references) the spectrum is roughly flat. Then from 0.08 to 0.035 (periods from 12.5 to 28.5 references) it decreases before again increasing sharply over the low frequencies (periods greater than 28.5 references).

The basic question is how to interpret the physical significance of this dip. We decided that a reasonable approach would be to find a method of synthesizing time sequences with similar spectra in the frequency domain. To do this, it is necessary to understand the statistical properties of the instruction stream, such as the distribution of inter-miss distances, etc.

6. Statistical properties of inter-miss distances

Figure 13 shows the histogram of the inter-miss distances for Trace 3, which spans a million storage references. The distribution of those distances falling between one and 16 references appears to be a classical "bell-shaped curve." The shape of the distribution curve over larger inter-miss distances is less obvious. The same histogram plotted on log-log paper (Fig. 14) shows, as expected, that when the distance becomes larger, the number of intervals in these ranges decreases. This means that our experimental estimate of the distribution becomes increasingly unreliable as the inter-miss distance increases.

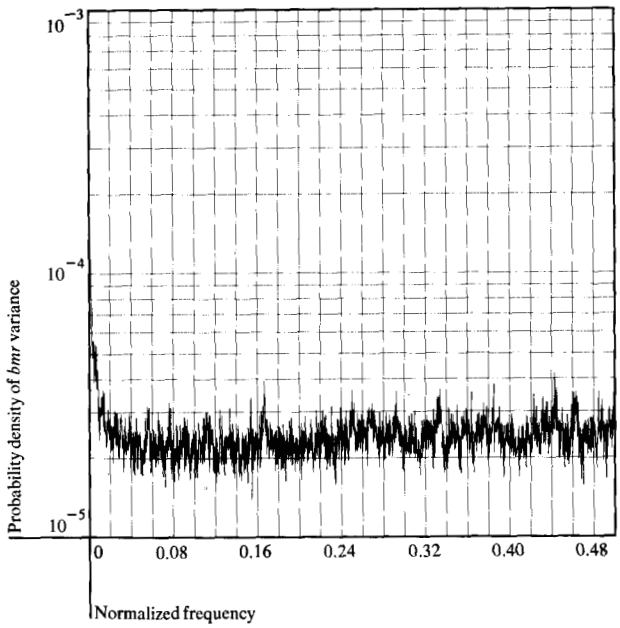

Figure 12 Miss spectrum of Trace 1 with all spectral lines filtered out.

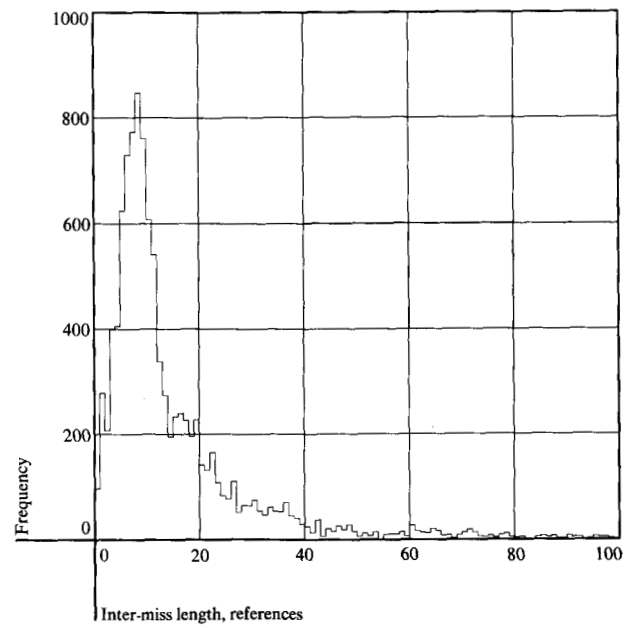

Figure 13 Histogram of inter-miss distances for Trace 3.

However, a reflection on the physical nature of the underlying phenomenon allows us to devise some reasonable hypothesis.

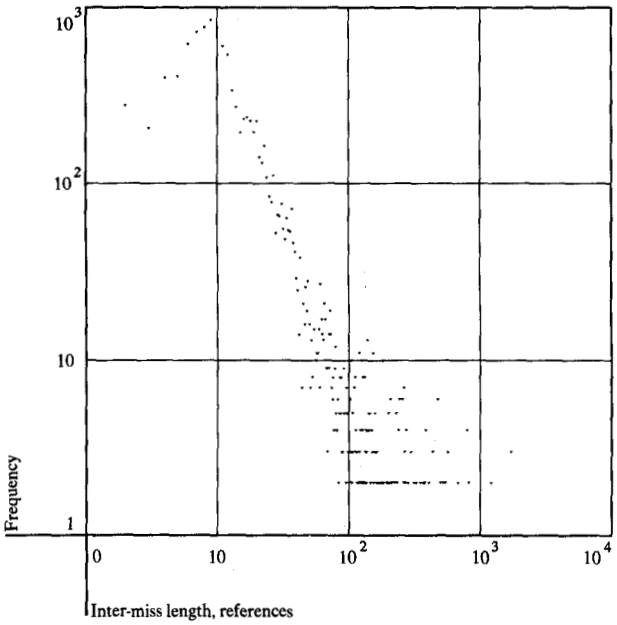

Figure 14 Distributions of inter-miss distances.

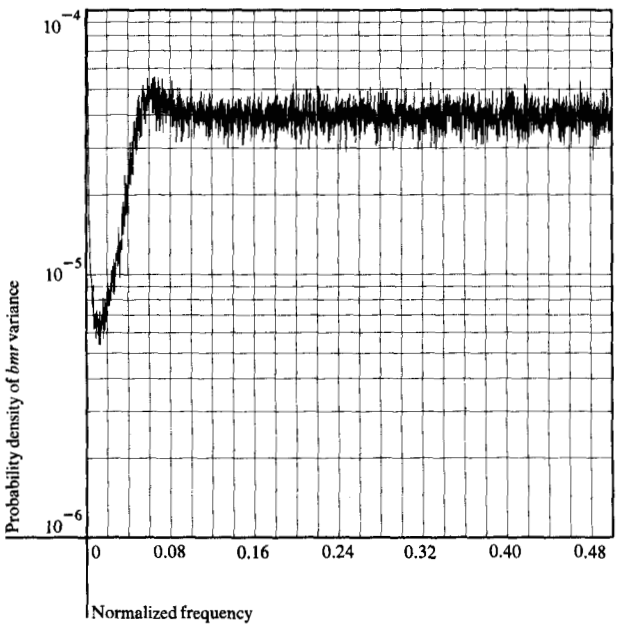

Figure 15 Synthetic miss sequence.

Underlying model

Our hypothesis is that the distribution of inter-miss distances is bi-modal. As a matter of fact, the underlying model could be as follows:

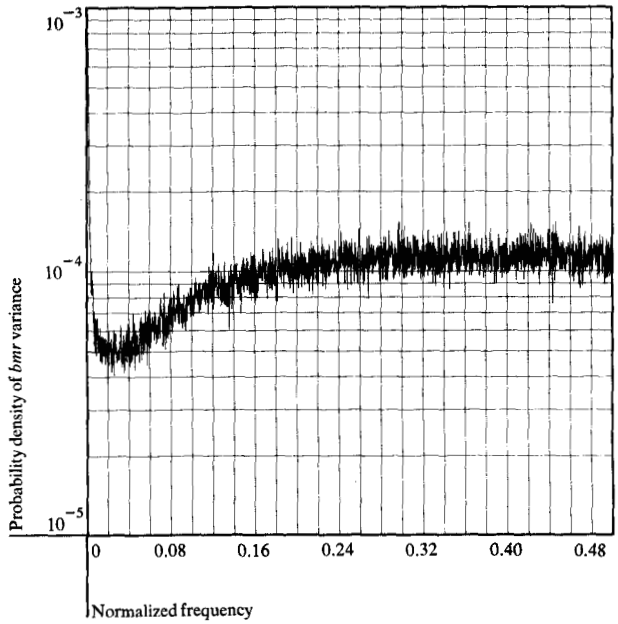

Figure 16 Synthetic miss sequence with average inter-miss distance within a burst of six references.

1. The small inter-miss distances are "bursts" of instruction fetch misses that occur after any sudden change in locality, *e.g.*, dispatches, subroutine calls.
2. The large inter-miss distances are "gaps" of long duration occurring between context switches, without any imbedded I-fetch misses.

Experiments

The high frequency cutoff

We construct a synthetic miss sequence $M(t)$ using 200 subsequences of exponentially distributed length with mean equal to 1025 references. These subsequences are shifted from one another by exponentially distributed shifts of mean 665 references. Each subsequence is populated by ones, with inter-one distance Erlang-distributed with 18 degrees of freedom, and thus with a mean of 18 references.

The total length of the resulting synthetic sequence is 360 000 references, which is the same as Trace 1. Its spectrum is shown in Fig. 15.

It is clear that this method produces a low frequency dip. However, this dip is narrower than the one in Fig. 6. The cutoff frequency, on the high frequency side of the dip, defined as the point where the curve cut the flat level portion of the spectrum, will be called the "high frequency cutoff." Its value is 0.055. This corresponds to a lag of about 18 references, which is the average inter-miss distance within a burst of misses.

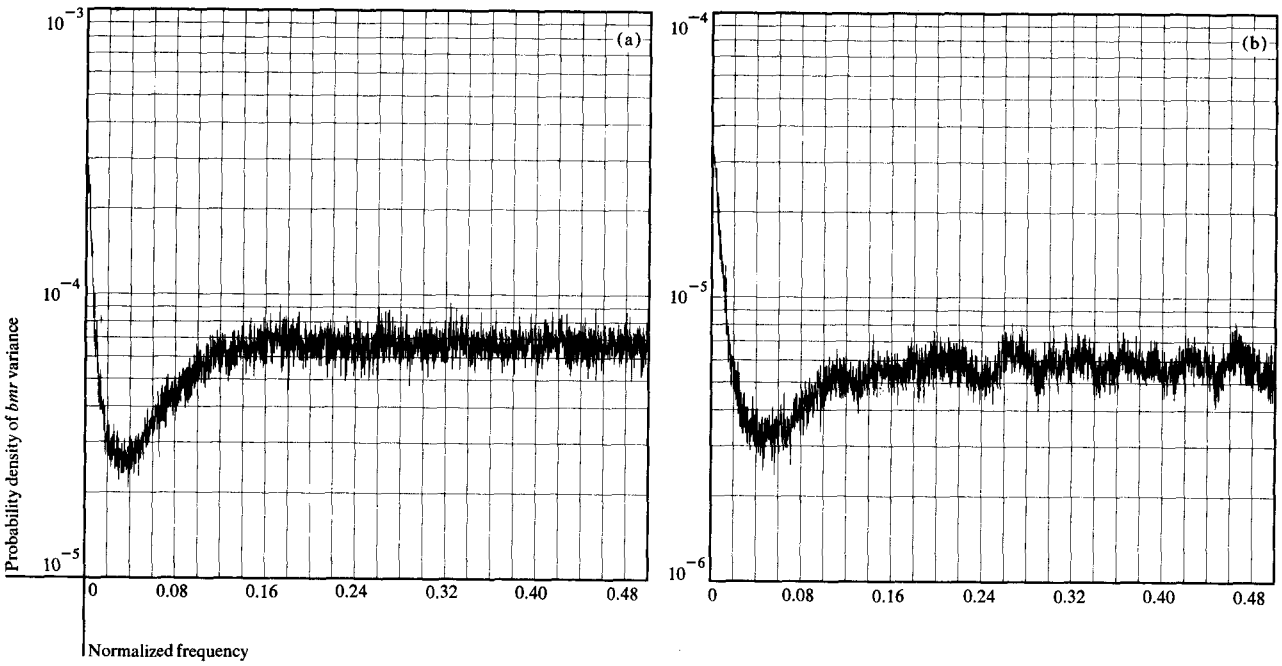

Figure 17 Spectrum of miss sequence with average length of a burst of misses equal to 100 references (a) and to 25 references (b).

To check this hypothesis, we built another synthetic miss sequence, identical in every respect to the first one, with the exception that the "average inter-miss distance within a burst" is six references.

The resulting spectrum is shown in Fig. 16, and indeed exhibits a dip with a high cutoff frequency value around 0.16, which corresponds to a lag of six references.

> Therefore, the high frequency cutoff value of the dip is the inverse of the mean value of the inter-miss distance within a burst of misses.

Let *HCO* be the notation for "high frequency cutoff" and *IMD* denote the mean inter-miss distance (expressed in references). Then

$$HCO = \frac{1}{\overline{IMD}} . \qquad (1)$$

The low frequency cutoff

Observe that in Fig. 6 the actual dip is more symmetrical than those synthesized up to now. Some more structure should therefore be added in such a way that it affects the extreme low frequency part of the spectrum.

We constructed a second set of synthetic sequences, keeping the average inter-miss distance constant (nine references) and varying the length of a burst of misses.

Figure 17(a) illustrates the spectrum of such a miss sequence with the average length of a burst of misses equal to a hundred references. The high frequency cutoff point did not move, but the low frequency cutoff point moved away from the ordinate axis to a value of 0.001, which corresponds to a lag of 100 references.

A similar experiment was conducted for a burst length of 25 references, producing the spectrum shown in Fig. 17(b). The low frequency cutoff point is equal to 0.02, which corresponds to a lag of 50 references.

At this point, we began to employ a more systematic approach involving three successive sets of experiments. In all of them, the average gap length was held constant at 665 references (with an exponential distribution).

For the first set of experiments, the inter-miss distance was fixed at an average of six references, while the average length of a burst was varied from 25 to 1025 references. The low frequency cutoff value was measured for each point and plotted against the average length of a burst.

This was repeated twice, changing the inter-miss distance first to nine and then to 18 references.

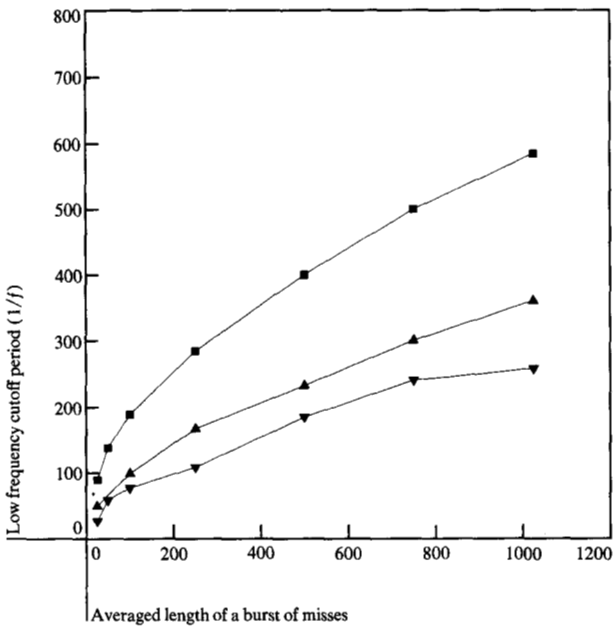

Figure 18 Family of curves.

A family of three curves, parameterized by the inter-miss distance value, is shown in Fig. 18. It is easy to recognize a set of parabolas, and indeed a curve fit provides us with the following three equations (with *LCO* denoting the low frequency cutoff):

$$\frac{1}{LCO} = 7.1 \times (\overline{burst})^{0.52},$$

$$\frac{1}{LCO} = 8.91 \times (\overline{burst})^{0.53},$$

$$\frac{1}{LCO} = 19.73 \times (\overline{burst})^{0.49}.$$

From this we deduce that a more general relationship for the low frequency cutoff value (*LCO*) is

$$\frac{1}{LCO} = \overline{IMD} \times \sqrt{(\overline{burst})}, \quad (2)$$

where *IMD* denotes the average inter-miss distance.

> In other words, the low frequency cutoff value is directly related to both the average length of a burst of misses and the average inter-miss distance within a burst.

Mean values

Mean burst length Now combining Eqs. (1) and (2), we can derive an expression for the mean length of a burst:

$$\overline{burst} = \left(\frac{HCO}{LCO}\right)^2. \quad (3)$$

Residual "noise level" The last variable in this model is the average value of the gap between successive bursts of instruction fetch misses. Keeping the average values of both the burst and the inter-miss distance constant at, respectively, 100 and 9 references, the value of the gap was increased from 100 references to 1025 references. We found that the values of the cutoff frequencies (both high and low) are not affected by this, except when the average gap value is equal to 100 references.

We can explain this anomaly. Consider that if the gap between the bursts of misses is too short, it becomes difficult to distinguish between "two successive misses within a burst" and "the last miss of a burst followed by the first miss of the following burst." This changes the mean values of the inter-miss distance and burst length because two bursts too close to one another will be treated as a single long burst with at least one long inter-miss distance occurring within the long burst.

Notwithstanding the above exception, the most noticeable effect of varying the mean gap length from 200 to 1000 references is a vertical downward shift of the spectrum. Using the flat level of the high frequency part as a reference level, it is possible to derive a linear relationship between the value of this reference "noise level" and the total number of misses.

Let *MI* be the number of misses, *NL* the reference noise level, and C_w a constant that depends on the type of data window used in smoothing the spectrum. (With smoothing in the frequency domain, $C_w = 1.0$, whereas using the Welch Segmentation Method gives $C_w = 1.66$.) We used regression analysis to determine the relationship between these variables, based on experimental data. This resulted in the following equation:

$$MI = 362\,000 \times 10^3 \times NL \times C_w.$$

Recall that all of the above synthetic sequences have a constant length of 360 000 references—*i.e.*, have the same length as Trace 1. Clearly this equation is parameterized in terms of the length (L) of the trace within the accuracy and size of our samples:

$$MI = L \times 10^3 \times NL \times C_w.$$

Then,

$$\frac{MI}{L} = bmr = 10^3 \times NL \times C_w. \quad (4)$$

> Thus, the level of the flat portion of the spectrum is proportional to the average buffer miss ratio of the instruction fetch trace.

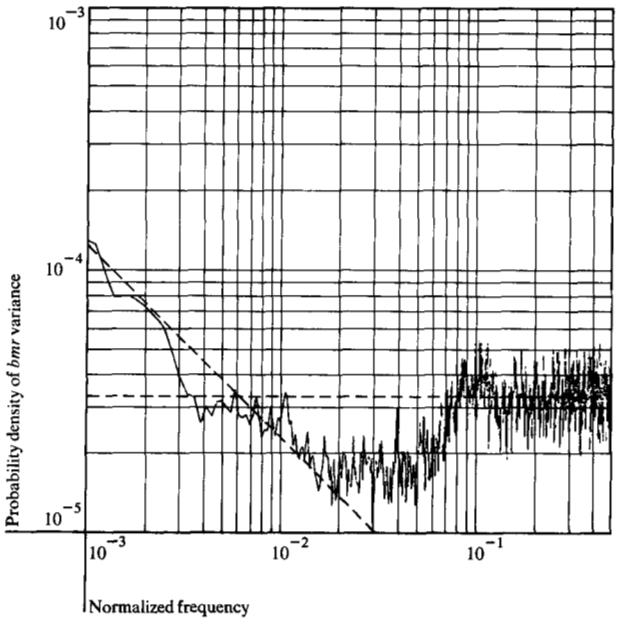

Figure 19 Instruction miss spectrum on a log-log scale.

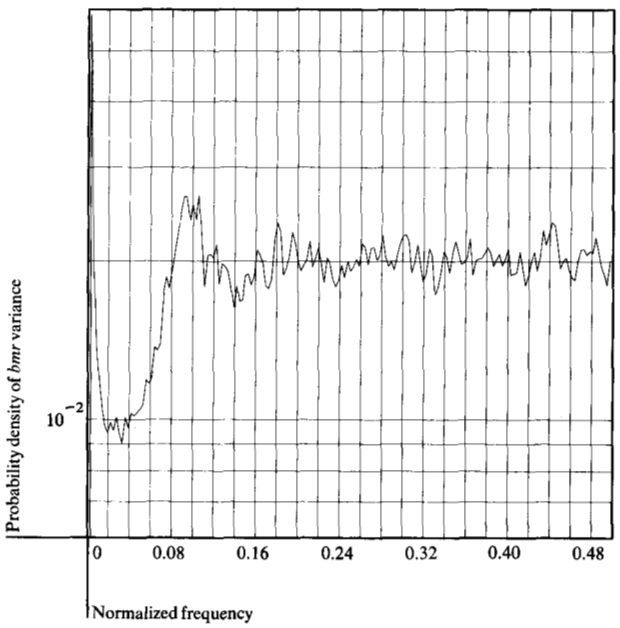

Figure 20 Recreated instruction miss spectrum of Fig. 6.

Mean gap length Now, if N_{burst} is the total number of bursts on a trace, then

$$L = (\overline{gap} + \overline{burst})N_{burst}.$$

Expressing the number of misses as the average length of a burst, divided by the average inter-miss distance within a burst, plus a corrective factor for gaps, we have

$$MI = \left(\frac{\overline{burst}}{\overline{IMD}} + 2\right)N_{burst}.$$

Now we have four equations which, given the spectrum of the instruction fetch miss sequence, define the main mean values characterizing the structure of an instruction fetch miss sequence. Recalling that *LCO* and *HCO* denote, respectively, low cutoff frequency and high cutoff frequency, it is possible to successively compute

$$\overline{burst} = \left(\frac{HCO}{LCO}\right)^2; \text{ then,}$$

$$N_{burst} = \frac{L \times NL \times C_w \times 10^3}{2 + \frac{1}{\overline{IMD}} \times \left(\frac{HCO}{LCO}\right)^2}, \text{ and}$$

$$\overline{gap} = \frac{L}{N_{burst}} - \overline{burst}. \quad (5)$$

- *Model validation*

If we go back to the original instruction fetch miss sequence of Trace 1 and its spectrum (Fig. 6), we can apply the above equations to a real case. For ease of measurement, Fig. 6 has been redrawn on a log-log scale (Fig. 19) to improve resolution in the low frequency range.

Using this graph, we find the level of the flat portion of the spectrum, which we called the noise level, to be equal to 3.3×10^{-5}.

This fixes the low cutoff frequency value at 0.006 and the high cutoff frequency at 0.08. Therefore,

$$IMD = \frac{1}{0.08} = 12.5 \text{ references and}$$

$$\overline{burst} = \left(\frac{0.08}{0.006}\right)^2 = 178 \text{ references; then,}$$

$$N_{burst} = \frac{3.3 \times 10^{-5} \times 10^3 \times 360\,000}{2 + 0.08 \times 178} = 732 \text{ and}$$

$$\overline{gap} = \frac{360\,000}{732} - 178 = 313 \text{ references.}$$

It is interesting to use these values to see to what extent the spectrum shown in Fig. 6 can be recreated by our model. The result can be seen on Fig. 20, and its shape reasonably validates this analytical formulation.

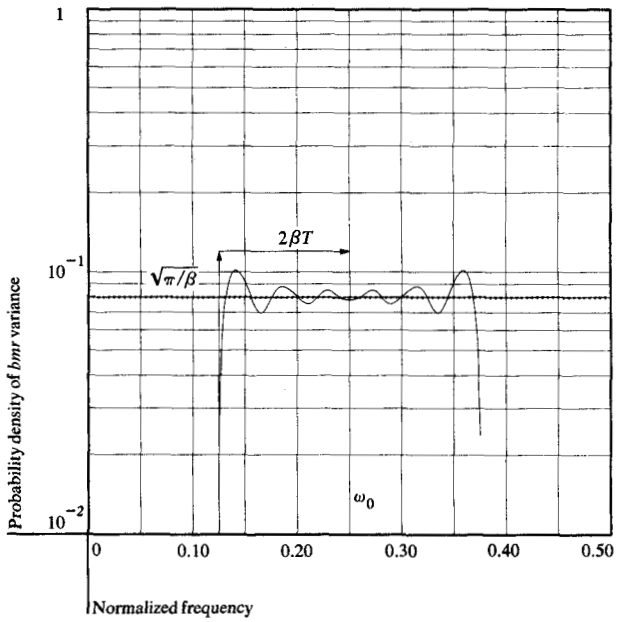

Figure 21 Spectrum of frequency modulated pulse.

● *Frequency modulation*

The model described above can be given another interpretation which might shed a different light upon the observed phenomenon.

The spectrum in Fig. 6 can be thought of as being made of two different parts:

1. A low frequency "bump" or spike of the $(\sin x)/x$ type. This is the transform of a burst of misses, perceived as a deformed rectangular signal. And we did indeed find in our previous experiences that the width of this $(\sin x)/x$ function was most sensitive to the length of a burst.
2. A high frequency semi-rectangular part, starting at the bottom of the dip. A closer look shows that this curve has the classical shape of a damped oscillation.

Now, if p_T is a rectangular pulse of duration $2T$, one may show that the transform of the frequency-modulated signal,

$p_T(t)e^{j\beta t^2}$, is

$$F(\omega)\left[K\left(\frac{2\beta T - \omega}{\sqrt{2\beta\pi}}\right) + K\left(\frac{2\beta T - \omega}{\sqrt{2\beta\pi}}\right)\right], \qquad (6)$$

where K denotes the Fresnel integral

$$K(x) = \int_0^x e^{\,j\pi\tau^2/2}\, d\tau \text{ and}$$

$$F(\omega) = \sqrt{\frac{\pi}{2\beta}}\, e^{-j\omega^2/4\beta}.$$

Figure 21 displays the shape of the spectrum obtained from this equation. It has a semi-rectangular form, with a superimposed damped oscillation, and is centered around a carrier frequency ω_0, with a maximum frequency excursion of $2\beta T$.

This is a close analog to the spectrum we are studying, at least over the higher frequencies.

We do not claim that these above equations represent an exact mathematical model of the spectrum of the observed phenomenon. In general, it is very difficult to come up with a closed form for the spectrum of a complex signal that is itself frequency modulated by another complex, and in this case stochastic, signal.

Thus, with the understanding that an expression for the spectrum would involve either Bessel functions or Fresnel functions, we experimentally fitted the spectrum of the instruction fetch misses of Trace 3 (Model 168, 32K cache) with a combination of Fresnel functions similar to expression (6).

Given the computational package available to us, it was easier for us to compute the Fresnel function $K(x)$ defined as

$$K(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \frac{e^{it}}{\sqrt{t}}\, dt.$$

Then, if f denotes the normalized frequency and b a constant, it is possible to fit the following curve on the high frequency portion of the Trace 3 spectrum in Fig. 22:

$$F_1(f) = b\left[K\left(\frac{200f - 12}{3}\right) + K\left(\frac{113 - 200f}{3}\right)\right]$$

with

$b = 0.04$.

This formulation is analogous to expression (6) but involves a $K(x)$ Fresnel function defined slightly differently from the one involved in expression (6).

Further, it is easy to fit a sin x/x function to the low frequency (close to frequency zero) portion of the spectrum. The specific formulation of this function is

$$F_2(f) = 330 \times \frac{\sin(2\pi f \times 330)}{2\pi f \times 330}.$$

But now the linear combination of both F_1 and F_2 is a reasonable match for the overall shape of the instruction fetch miss spectrum shown in Fig. 6. The generalization, therefore, is that all such spectra can be *decomposed* into two distinct parts:

1. A low frequency sin x/x spike, caused by the perception of the envelope of a burst of misses to be a deformed rectangular signal of substantial length (330 references in this case).
2. A high frequency step function, characteristic of frequency modulated signals, with a damped oscillation which can be accounted for by a Fresnel function. This captures our intuitive perception of misses within a burst as a set of frequency modulated pulses.

Hence, the high frequency cutoff is further understood to be a lower bound for the frequency of sequential instruction-fetch misses, after a change of locality has occurred.

7. Physical interpretation

The above analysis shows that most of the information about the structure of the instruction fetch miss sequence can be read off its spectrum.

It is now time to close the loop and relate these data to the physical parameters of either the cache or program behavior.

- *Line size and mean burst length*

By keeping the size of the cache constant at 64K, and varying the size of the line from 32 bytes to 128 bytes, we find the following two relations for the Trace 3:

$$\overline{IMD} = 0.39 \times L_s \text{ and} \tag{7}$$

$$\frac{1}{LCO} = 7.1 \times L_s, \tag{8}$$

where L_s denotes the line size of the cache.

Both variables, $\overline{IMD}$ and LCO, can be substituted in Eq. (2), yielding

$$\overline{burst} = \left(\frac{7.1L_s}{0.39L_s}\right)^2 \text{ and}$$

$$\overline{burst} = 331 \text{ references.}$$

This result is obviously independent of the size of the line in the cache.

> The above analysis can be summed up by saying that while the inter-miss distance is proportional to the size of the cache line, the length of a burst of misses following a context switch is independent of this very line size.

- *Mean burst length and cache size*

Now keeping the size of the cache line constant at 128 bytes, we vary the size of the cache from 32K to 256K. For each value we compute the mean length of the burst using Eq. (3). These computations are made for four points (32K, 64K, 128K, 256K) and a curve fit on the results, which provides us with the following equation relating the size of the cache to the length of the burst. Let C_s be the size of the cache, expressed in K (1024 bytes); then,

$$\overline{burst} = \frac{20\,000}{C_s + 4}, \tag{9}$$

with C_s denoting the cache size.

> Hence the average length of a burst is inversely proportional to the size of the cache, and is not dependent on the size of the line.

Obviously the constant 20 000 should relate to the instruction stream itself, but further investigations are necessary to prove this point.

8. Conclusions

In summary, this dip is indicative of two distinct distributions of straight line instruction misses: (1) the distribution of bursts themselves; and (2) the distribution of individual straight line instruction misses within a burst (Fig. 22). Note that if the bursts are widely separated, yet have a high miss density within each burst, then these two distributions will have disjoint spectra, leading to the peculiar dip noted above. The time domain interpretation, then, is

- Each subroutine call or context switch triggers a burst of instruction fetches, with subsequent misses exponentially distributed within that burst.
- Then, for a while (gap), there are no instruction fetch misses. The length of these gaps is also an exponential distribution.
- Then another subroutine is called and triggers a new burst of instruction fetch misses, and the process repeats itself.

A simple relationship exists between the average length of the burst, the inter-miss distance within a burst, and the low frequency cutoff value. In addition the high frequency cutoff value is the inverse of the inter-miss distance, and therefore the arrival rate of misses, if references are to be taken as a measure of time. Thus, the high frequency cutoff can be interpreted as a lower bound

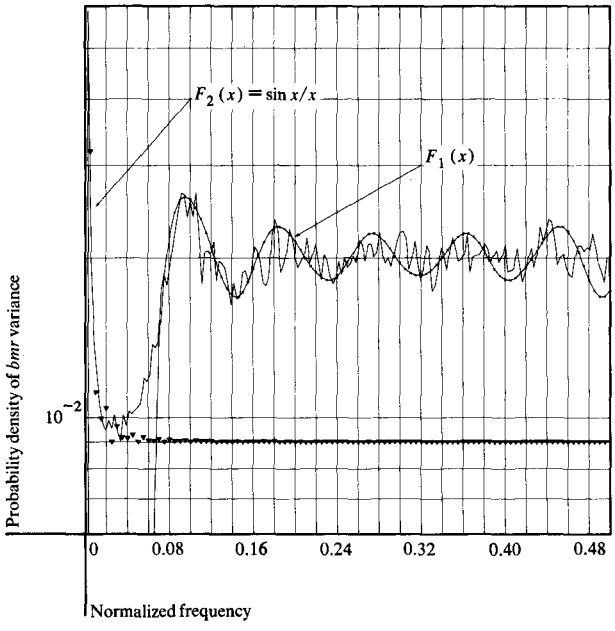

Figure 22 Miss spectrum of Trace 3.

for the frequency of straight line instruction fetch (SLIF) misses, when a change of locality occurs. In other words the sequential misses will be more frequent in a context switch. This also explains why the density of SLIF misses within a burst is proportional to the line size (Eq. 7).

This last variable is proportional to the line size in the cache. The mean length of a burst is inversely proportional to the size of the cache and independent of the size of the line.

Furthermore, we have demonstrated in the first part of this paper the existence of periodic data misses, caused by tight loops into regular data structures, and we provided a tool, based on filtering of miss sequences, which identifies easily and very accurately these pieces of code.

In general, we observed that the main mechanisms behind cache misses are well identified, or easily identifiable, software events.

Acknowledgment

The authors wish to acknowledge the invaluable aid of Dr. Shmuel Winograd in the development of this paper. Without his help, our analysis would have been crippled by numerous errors and omissions of understanding. We also would like to thank Dr. J. W. Cooley and Dr. P. D. Welch for their time and efforts expanded on our behalf. Further, we could never even have begun our investigations had not the Poughkeepsie Laboratory shared with us their special expertise, trace tapes, and utility programs. Indeed, we interacted closely with both the Poughkeepsie Timer and Performance groups and with the developers of IMS at Santa Teresa.

Appendix A

Let us first construct a building block sequence, $m(t)$, of period 8, *i.e.*,

$m(t) =$ "100000001000000010 · · ·"

and of length 50 000.

The discrete Fourier transform of this periodic function is given by

$$X(k) = \sum_{t=0}^{N-1} m(t) e^{-2\pi i \frac{t}{N} k},$$

with $t = 0, 1, 2, \cdots, N-1$. One property of this binary sequence is that each $m(t)$ value is zero except for those values of t which are multiples of 8, in which case it is equal to one. Then let t be $8u$ and let M be $N/8$; the above equation can be rewritten as

$$X(k) = \sum_{u=0}^{\frac{N}{8}-1} e^{-2\pi i \frac{8u}{8M} k}$$

and

$$X(k) = \sum_{u=0}^{M-1} e^{2\pi i \frac{u}{M} k}. \tag{A1}$$

Two cases are now possible depending on whether k is a multiple of M. In this example, M is equal to 6250 references. Then in the case where $k = n \times 6250$, *i.e.*, $k = 0, 6250, 12\,500, 18\,750$, the exponent of the exponential is equal to a multiple of 2π, and the exponential is therefore equal to one. Then $X(k)$ is a sum of M ones and thus equal to M, which happens to be the number of periodic misses. Hence, we have sharp spikes of equal height at those values of k which would correspond to normalized frequencies $k/50\,000 = 0, 0.125, 0.250, 0.375$. (This can be read as 0/8, 1/8, 2/8, and 3/8.)

Now, in the other case where k is not a multiple of 6250, the equation giving $X(k)$ can be thought of as being the transform of the binary sequence of length 6250:

$m'(u) =$ "11111111111 · · ·"

Equation (A1) then can be rewritten as

$$X(k) = 1 + e^{-2\pi i \frac{1}{M}} + e^{-2\pi i \frac{2}{M}} + \cdots + e^{-2\pi i \frac{M-1}{M}},$$

which is of the form

$$X(k) = 1 + z + z^2 + z^3 + \cdots + z^{M-1}.$$

The case where z was equal to one was treated above. Hence z is different from one, and $X(k)$ is equal to

$$X(k) = \frac{1 - e^{-2\pi i \frac{M}{M}}}{1 - e^{-2\pi i \frac{1}{M}}}.$$

The numerator of this expression is obviously zero, and hence $X(k)$ is zero for any k which is not a multiple of 6250 references.

In summary, the transform of a binary sequence of length N, periodic with period p, is made of spikes at normalized frequencies: $0/p$, $1/p$, $2/p$, $\cdots$, $p - 1/p$. The value $1/p$ is called the fundamental frequency, and the others, the harmonics of such frequency. It is a property of the transform to be symmetric about frequency 0.5. Therefore, we usually compute the spectrum from frequency zero to frequency 0.5, and only half of those spikes are usually shown.

We next try to answer the following question: What will be the shape of the spectrum of a binary sequence made up of a small number of non-overlapping periodic subsequences, each shifted from one another by a random number of references?

We will use the fact that the Fourier transform is a linear operator. For the sake of simplicity, we assume that the first subsequence starts at the zero origin of time and is therefore identical to the one analyzed above. Then the resulting transform for this composite sequence will be the sum of the transforms of each subsequence. So we start with the transform of the first subsequence, *i.e.*, spikes at 0, 0.125, 0.250, and 0.375. [This spectrum will itself be convolved with a very narrow sin x/x function, coming from the finite length (50 000) of this subsequence, and therefore not much different from the spectrum computed above.]

To this transform one will add the transform of the second periodic subsequence. This second subsequence is identical to the first but shifted in the time domain by some number δ of references. The distance of the first point of the second subsequence, counted from the origin of time, is thus $N + \delta$, while the total length of the composite sequence is $2N + \delta$.

A theorem of the Fourier transform theory states that a shift of $N + \delta$, in the time domain, amounts to a multiplication of the transform by

$$e^{-i(N+\delta)2\pi \frac{k}{2N+\delta}}$$

Now p, denoting the period within the subsequence, spikes would appear for values of k equal to $r(2N + \delta)/p$, r being an integer with values 0, 1, 2, $\cdots$, $p - 1$. Therefore, one can rewrite the above expression as

$$e^{-ir(N+\delta)2\frac{\pi}{p}}.$$

Now, in general,

$$\delta = ap + q$$

and

$$e^{-ir(ap+q)2\pi/p} = e^{-2\pi ir\frac{q}{p}}.$$

This shows that the phase shift depends only on the ratio q/p and on the value of r. For $r = 0$, that is, for the dc term, the shift is null. For $r = 1$, which corresponds to the fundamental, the angle is $2\pi q/p$. Thus spikes will be unequally affected by the random phase shifts, and, bearing in mind that they were equal at the beginning, their final respective heights will be different once all transforms have been added to the first one.

References

1. J. S. Liptay, "Structural Aspects of the System/360 Model 85, The Cache," *IBM Syst. J.* **7**, 15–21 (1968).
2. C. J. Conti, "Concepts for Buffer Storage," *IEEE Computer Group News* **2**, 9–13 (1969).
3. J. Voldman and L. W. Hoevel, "The Fourier-Cache Connection," *Digest of Papers, Com-Con Spring of '81*, IEEE Order Number 341, New York, pp. 344–354.
4. J. R. Spirn, *Program Behavior: Models and Measurements*, Elsevier Scientific Publishing Co., Inc., New York, 1977.
5. P. A. W. Lewis and G. S. Shedler, "Empirically Derived Micromodels for Sequences of Page Exceptions," *IBM J. Res. Develop.* **17**, 86–100 (1973).
6. A. V. Oppenheim and R. W. Schafer, *Digital Signal Analysis*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1975.
7. G. M. Jenkins and D. G. Watts, *Spectral Analysis and its Applications*, Holden Day, San Francisco, CA, 1968.
8. P. Bloomfield, *Fourier Analysis of Time Series: An Introduction*, John Wiley & Sons, Inc., New York, 1976.
9. A. Papoulis, *Signal Analysis*, McGraw-Hill Book Co., Inc., New York, 1977.
10. E. Parzen, *Time Series Analysis Papers*, Holden Day, San Francisco, 1967.
11. J. W. Cooley, P. A. W. Lewis, and P. D. Welch, "The Fast Fourier Transform Algorithm and its Applications," *Research Report RC 1743*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1967.

Received July 7, 1980; revised May 15, 1981

The authors are located at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.